IN THE CLAIMS

Claim 1.   (Currently Amended)   A<u>n</u> apparatus comprising:

a first processor and a second processor each having a scoreboard and a decoder;

a plurality of memory devices coupled to the first processor and the second processor;

a first buffer coupled to the first processor and the second processor, the first buffer being a register buffer;

a second buffer coupled to the first processor and the second processor, the second buffer being a trace buffer; and

a plurality of memory instruction buffers coupled to the first processor and the second processor;

wherein the first processor and the second processor perform single threaded applications using multithreading resources, and the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction<u>, and said single threaded application is not converted to an explicit multiple-thread</u>.

Claim 2.   (Previously Presented)   The apparatus of claim 1, wherein the memory devices comprise a plurality of cache devices.

Claim 3.   (Original)   The apparatus of claim 1, wherein the first processor is coupled to at least one of a plurality of zero level (L0) data cache devices and at least one of a plurality of L0 instruction cache devices, and the second processor is coupled to at least one of the plurality of L0 data cache devices and at least one of the plurality of L0 instruction cache devices.

Claim 4.   (Previously Presented)   The apparatus of claim 3, wherein each of the plurality of L0 data cache devices store exact copies of store instruction data.

Claim 5.   (Original)   The apparatus of claim 1, wherein the plurality of memory instruction buffers includes at least one store forwarding buffer and at least one load-ordering buffer.

Claim 6.   (Original)   The apparatus of claim 5, the at least one store forwarding buffer comprising a structure having a plurality of entries, each of the plurality of entries having a tag portion, a validity portion, a data portion, a store instruction identification (ID) portion, and a thread ID portion.

Claim 7.   (Original)   The apparatus of claim 6, the at least one load ordering buffer comprising a structure having a plurality of entries, each of the plurality of entries having a tag portion, an entry validity portion, a load identification (ID) portion, and a load thread ID portion.

Claim 8   (Canceled)

Claim 9.   (Previously Presented)   The apparatus of claim 1, wherein the trace buffer is a circular buffer.

Claim 10.   (Original)   The apparatus of claim 1, the register buffer comprising an integer register buffer and a predicate register buffer.

Claim 11.   (Currently Amended)   A method comprising:
         executing a plurality of instructions in a ~~first~~ single thread by a first processor;
         executing said plurality of instructions in the ~~first~~ single thread by a second processor as directed by the first processor, the second processor executing said plurality of instructions ahead of the first processor to avoid misprediction;
         tracking at least one register that is one of loaded from a register file buffer, and written by said second processor, said tracking executed by said second processor,
         transmitting control flow information from the second processor to the first processor, the first processor avoiding branch prediction by receiving the control flow information;
         transmitting results from the second processor to the first processor, the first

processor avoiding executing a portion of instructions by committing the results of the portion of instructions into a register file from a first buffer, the first buffer being a trace buffer, and

clearing a store validity bit and setting a mispredicted bit in a load entry in the first buffer if a replayed store instruction has a matching store identification (ID) portion in a second buffer, the second buffer being a load buffer,
wherein the first processor and the second processor execute single threaded applications using multithreading resources, and said single thread is not converted to an explicit multiple-thread.

Claim 12.   (Cancelled)

Claim 13.   (Previously Presented)   The method of claim 11, further including:
duplicating memory information in separate memory devices for independent access by the first processor and the second processor.

Claim 14.   (Cancelled)

Claim 15.   (Previously Presented)   The method of claim 11, further including:
setting a store validity bit if a store instruction that is not replayed matches a store identification (ID) portion in a load buffer.

Claim 16.   (Previously Presented)   The method of claim 11, further including:
flushing a pipeline, setting a mispredicted bit in a load entry in the trace buffer and restarting a load instruction if one of the load is not replayed and does not match a tag portion in a load buffer, and the load instruction matches the tag portion in the load buffer while a store valid bit is not set.

Claim 17.   (Previously Presented)   The method of claim 11, further including:
executing a replay mode at a first instruction of a speculative thread.

Claim 18.   (Previously Presented)   The method of claim 11, further including:
supplying names from the trace buffer to preclude register renaming;

issuing all instructions up to a next replayed instruction including dependent instructions;

issuing instructions that are not replayed as no-operation (NOPs) instructions;

issuing all load instructions and store instructions to memory;

committing non-replayed instructions from the trace buffer to the register file.

Claim 19.   (Previously Presented)   The method of claim 11, further including:

clearing a valid bit in an entry in a load buffer if the load entry is retired.

Claim 20.   (Currently Amended)   An apparatus comprising a machine-readable medium containing instructions which, when executed by a machine, cause the machine to perform operations comprising:

executing a ~~first~~ single thread from a first processor;

executing said ~~first~~ single thread from a second processor as directed by the first processor, the second processor executing instructions ahead of the first processor to avoid misprediction;

tracking at least one register that is one of loaded from a first buffer, and written by said second processor, said tracking executed by said second processor, the first buffer being a register file buffer, and

clearing a store validity bit and setting a mispredicted bit in a load entry in a second buffer if a replayed store instruction has a matching store identification (ID) portion, the second buffer being a trace buffer, <u>wherein the first processor and the second processor execute single threaded applications using multithreading resources, and said single thread is not converted to an explicit multiple-thread.</u>

Claim 21.   (Original)   The apparatus of claim 20, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

transmitting control flow information from the second processor to the first processor, the first processor avoiding branch prediction by receiving the control flow information.

Claim 22. (Original) The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

duplicating memory information in separate memory devices for independent access by the first processor and the second processor.

Claim 23. (Cancelled)

Claim 24. (Original) The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

setting a store validity bit if a store instruction that is not replayed matches a store identification (ID) portion.

Claim 25. (Previously Presented) The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

flushing a pipeline, setting a mispredicted bit in a load entry in the second buffer and restarting a load instruction if one of the load is not replayed and does not match a tag portion in a load buffer, and the load instruction matches the tag portion in the load buffer while a store valid bit is not set.

Claim 26. (Previously Presented) The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

executing a replay mode at a first instruction of a speculative thread;

terminating the replay mode and the execution of the speculative thread if a partition in the second buffer is approaching an empty state.

Claim 27. (Previously Presented) The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

supplying names from the second buffer to preclude register renaming;

issuing all instructions up to a next replayed instruction including dependent instructions;

issuing instructions that are not replayed as no-operation (NOPs) instructions;

issuing all load instructions and store instructions to memory;

committing non-replayed instructions from the second buffer to a register file.

Claim 28.   (Original)   The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

clearing a valid bit in an entry in a load buffer if the load entry is retired.

Claim 29.   (Currently Amended)   A system comprising:

a first processor and a second processor each having a scoreboard and a decoder;

a bus coupled to the first processor and the second processor;

a main memory coupled to the bus;

a plurality of local memory devices coupled to the first processor and the second processor;

a first buffer coupled to the first processor and the second processor, the first buffer being a register buffer;

a second buffer coupled to the first processor and the second processor, the second buffer being a trace buffer; and

a plurality of memory instruction buffers coupled to the first processor and the second processor,
wherein the first processor and the second processor perform single threaded applications using multithreading resources, ~~and~~ the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, <u>and said single thread is not converted to an explicit multiple-thread</u>.

Claim 30.   (Original)   The system of claim 29, the local memory devices comprise a plurality of cache devices.

Claim 31. (Original) The system of claim 30, the first processor is coupled to at least one of a plurality of zero level (L0) data cache devices and at least one of a plurality of L0 instruction cache devices, and the second processor is coupled to at least one of the plurality of L0 data cache devices and at least one of the plurality of L0 instruction cache devices.

Claim 32. (Previously Presented) The system of claim 31, wherein each of the plurality of L0 data cache devices store exact copies of store instruction data.

Claim 33. (Original) The system of claim 31, the first processor and the second processor each sharing a first level (L1) cache device and a second level (L2) cache device.

Claim 34. (Original) The system of claim 29, wherein the plurality of memory instruction buffers includes at least one store forwarding buffer and at least one load ordering buffer.

Claim 35. (Original) The system of claim 34, the at least one store forwarding buffer including a structure having a plurality of entries, each of the plurality of entries having a tag portion, a validity portion, a data portion, a store instruction identification (ID) portion, and a thread ID portion.